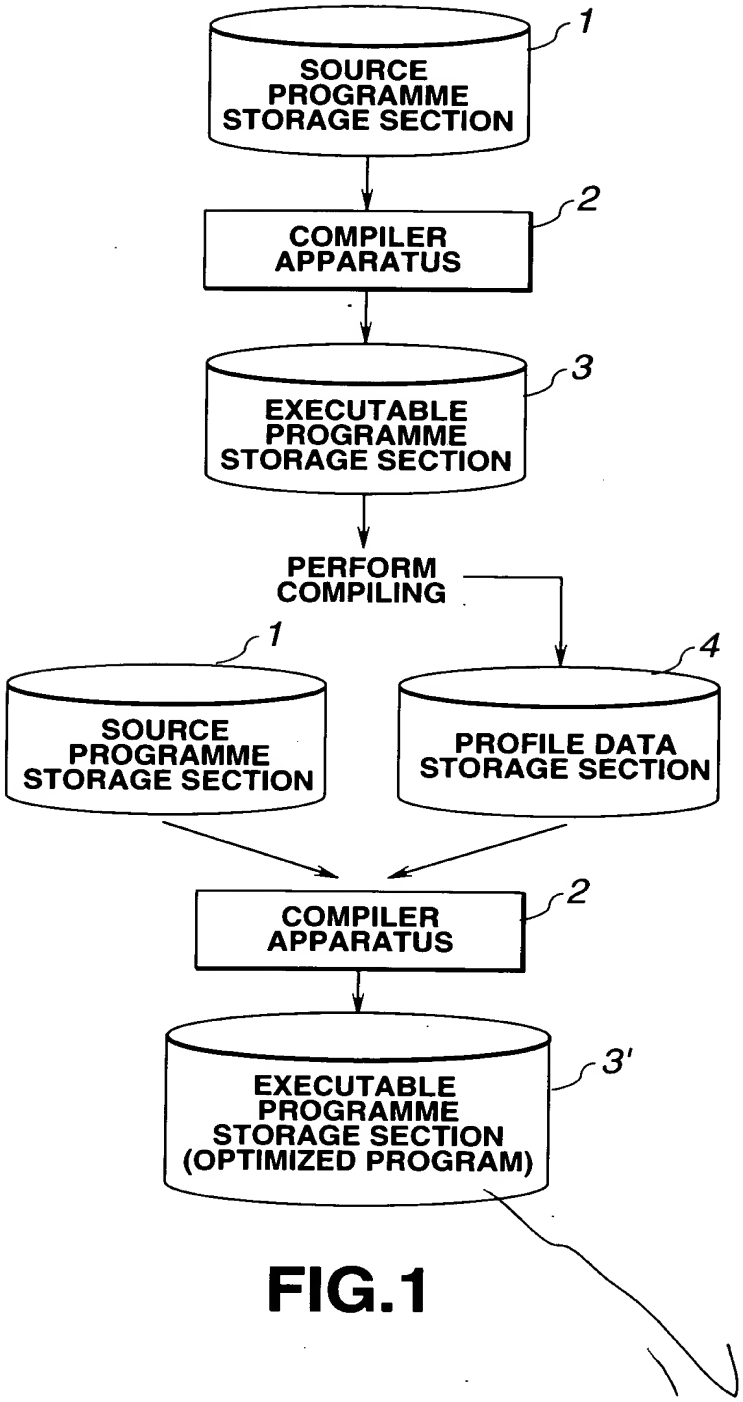
1
SOURCE PROGRAMME STORAGE SECTION
2
COMPILER APPARATUS
3
EXECUTABLE PROGRAMME STORAGE SECTION
PERFORM COMPILING
1
SOURCE PROGRAMME STORAGE SECTION
4
PROFILE DATA STORAGE SECTION
2
COMPILER APPARATUS
3'
EXECUTABLE PROGRAMME STORAGE SECTION (OPTIMIZED PROGRAM)

FIG.1

MAR 2000 PCT/PTO

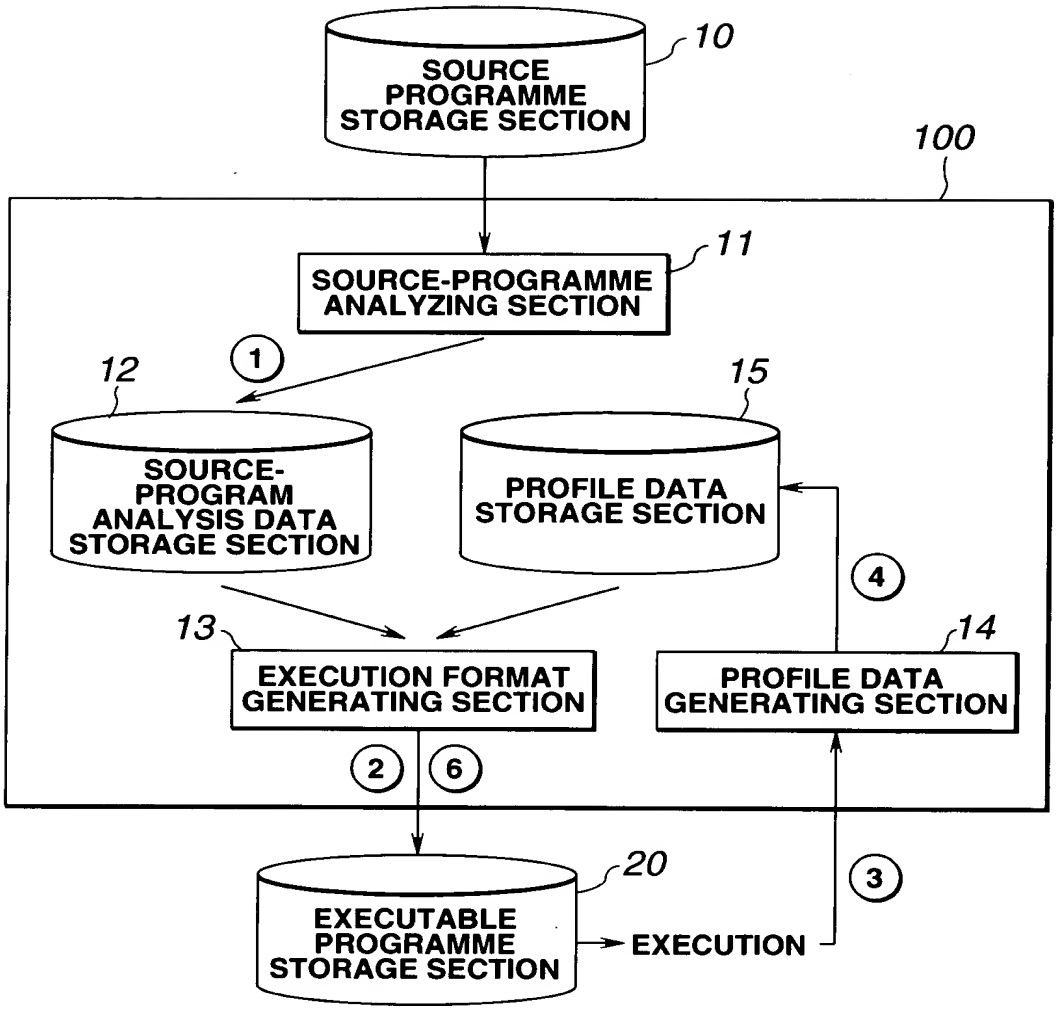

FIG.2

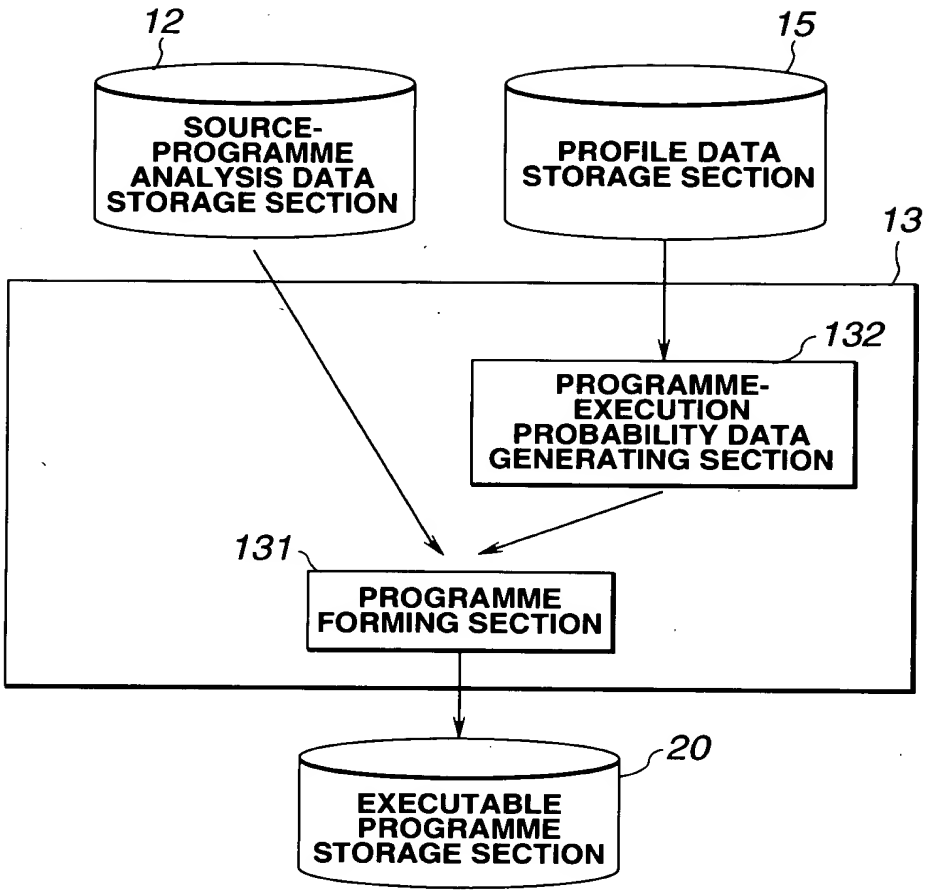
12
SOURCE-PROGRAMME ANALYSIS DATA STORAGE SECTION
15
PROFILE DATA STORAGE SECTION
13
132
PROGRAMME-EXECUTION PROBABILITY DATA GENERATING SECTION
131
PROGRAMME FORMING SECTION
20
EXECUTABLE PROGRAMME STORAGE SECTION

FIG.3

09/486865

Rec'd PCT/PTO 02 MAR 2000

EXAMPLE OF A SOURCE PROGRAMME

```
if (a>b)        then c = a ;
                else c = b ;

d = c  c ;
```

FIG.4

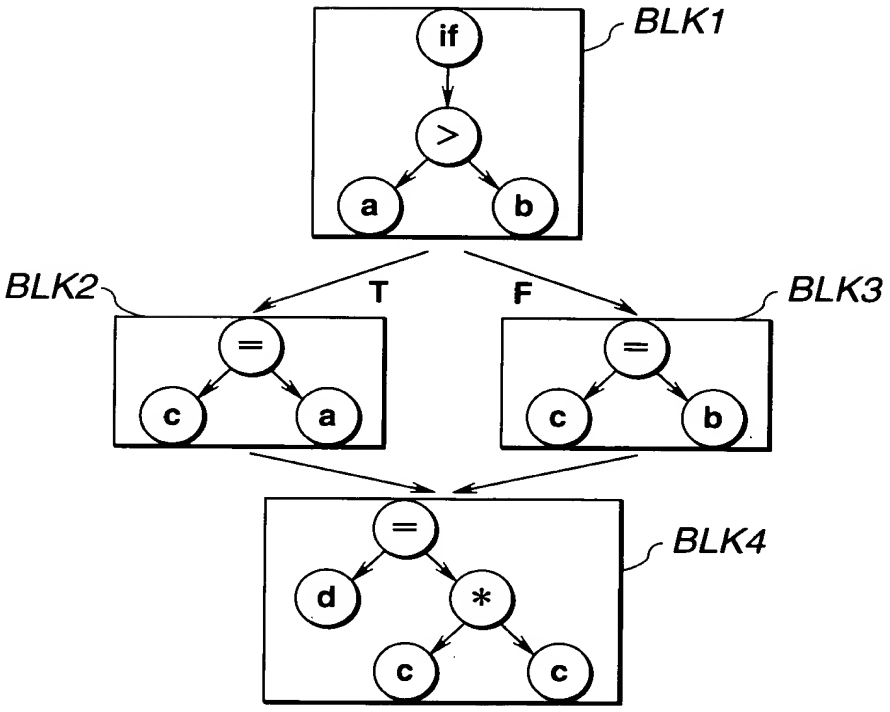

FIG.5

ASSEMBLY CODE

```
. text
block_1 :
        ld      r1, [a]          ; r1<-a
        ld      r2, [b]          ; r2<-b
        cmp     r3, r1, r2       ; r3<-r1 cmp r2
        ble     r3, block_3
block_2 :
        ld      r4, [a]          ; r4<-a
        st      r4, [c]          ; c<-r4
        jmp     block_4
block_3 :
        ld      r5, [b]          ; r5<-b
        st      r5, [c]          ; c<-r5
        jmp     block_4
block_4 :
        ld      r6, [c]          ; r6<-c
        ld      r7, [c]          ; r7<-c
        mul     r8, r6, r7       ; r8<-r6 * r7
        st      r8, [d]          ; d<-r8
```

FIG.6

PROFILE DATA

TIME	DETAILED DATA	
. . .		
10050:	PC	block_1
	load address	a
10051:	load address	b
10052:	compare	
10053:	branch less or equal	block_3
10054:	PC	block_3
	load address	b
10055:	store address	c
10056:	jump	block_4
10057:	PC	block_4
	load address	c
10058:	load address	c
10059:	mul	
10080:	store address	d
. . .		

FIG.7

PROBABILITY DATA

LABEL NAME	EXECUTION/ NUMBER OF ACCESSES
. . .	
block_1 :	100
block_2 :	10
block_3 :	90
block_4 :	100
. . .	
a load:	110
a store:	0
b load:	190
b store:	0
c load:	200
c store:	100
d load:	0
d store:	100
. . .	

FIG.8

```
. text
block_1 :
          ld      r1, [a]          ; r1<-a
          ld      r2, [b]          ; r2<-b
          mv      r5, r2           ; r5<-r2
          cmp     r3, r1, r2       ; r3<-r1 cmp r2
          ble     r3, block_3
block_2 :
          ld      r4, [a]          ; r4<-a
          st      r4, [c]          ; c<-r4
          jmp     block_4
block_3 :
          st      r5, [c]          ; c<-r5
          mul     r8, r5, r5       ; r8<-r5 * r5
          st      r8, [d]          ; d<-r8
          jmp     block_5
block_4 :
          ld      r6, [c]          ; r6<-c
          ld      r7, [c]          ; r7<-c
          mul     r8, r6, r7       ; r8<-r6 * r7
          st      r8, [d]          ; d<-r8
```

FIG.9

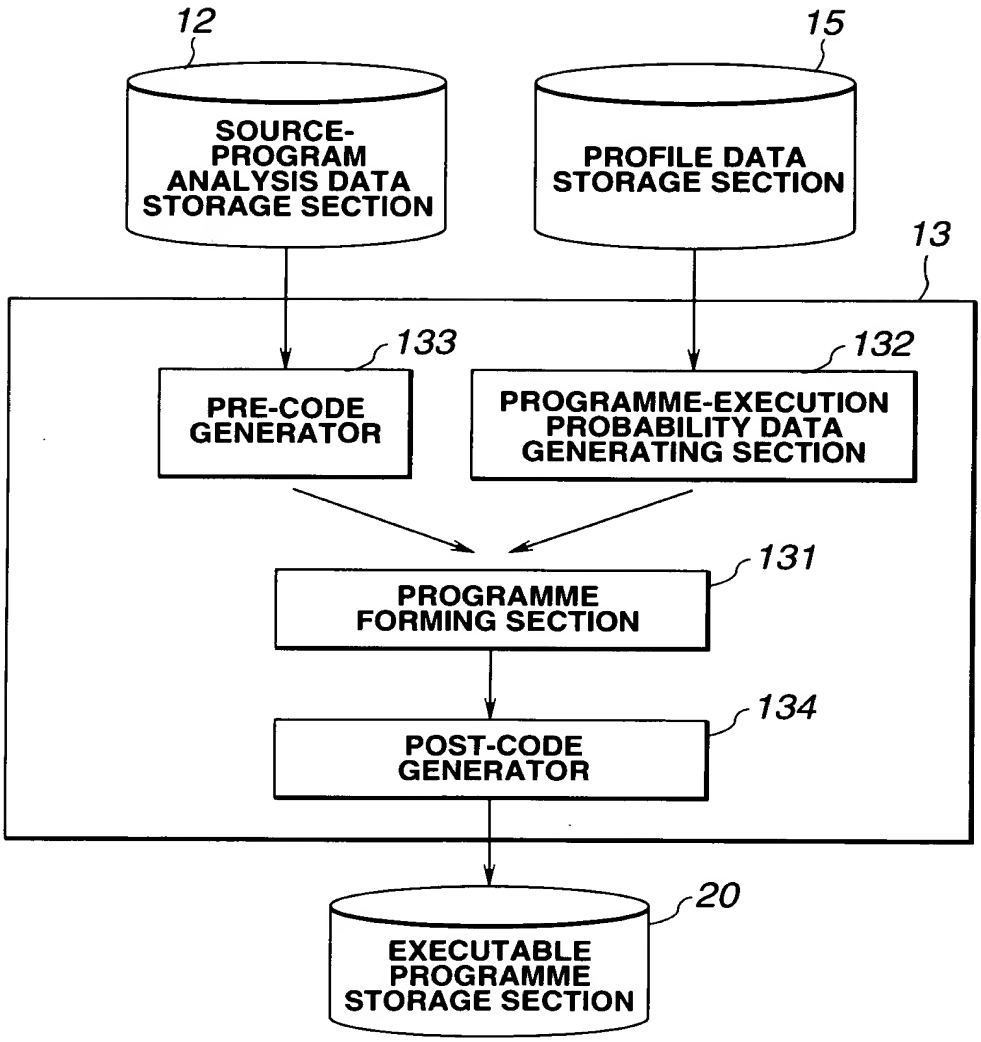
12
SOURCE-PROGRAM ANALYSIS DATA STORAGE SECTION
15
PROFILE DATA STORAGE SECTION
13
133
PRE-CODE GENERATOR
132
PROGRAMME-EXECUTION PROBABILITY DATA GENERATING SECTION
131
PROGRAMME FORMING SECTION
134
POST-CODE GENERATOR
20
EXECUTABLE PROGRAMME STORAGE SECTION

FIG.10

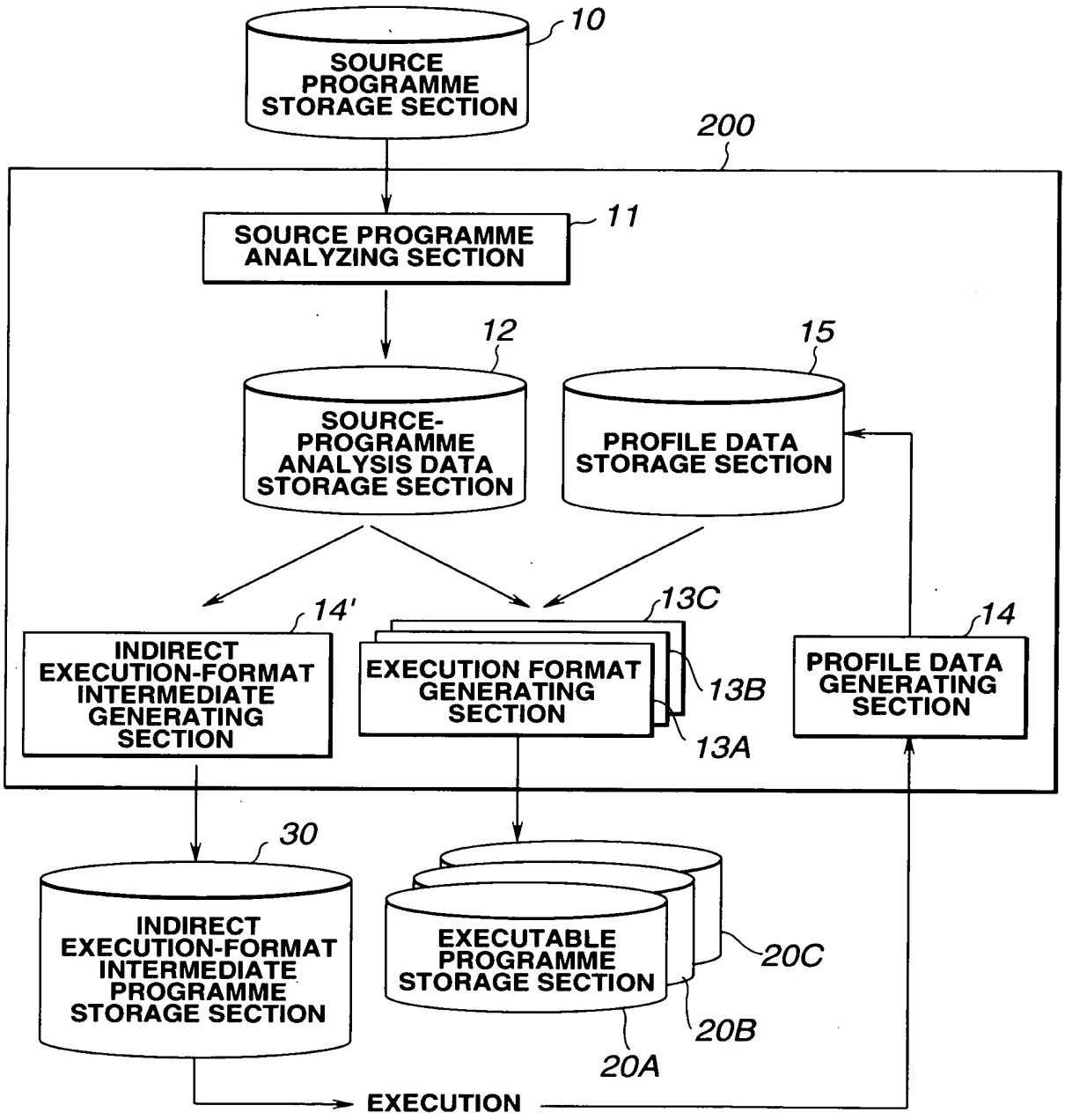
10
SOURCE PROGRAMME STORAGE SECTION
200
11
SOURCE PROGRAMME ANALYZING SECTION
12
SOURCE-PROGRAMME ANALYSIS DATA STORAGE SECTION
15
PROFILE DATA STORAGE SECTION
14'
INDIRECT EXECUTION-FORMAT INTERMEDIATE GENERATING SECTION
13C
EXECUTION FORMAT GENERATING SECTION
13B
13A
14
PROFILE DATA GENERATING SECTION
30
INDIRECT EXECUTION-FORMAT INTERMEDIATE PROGRAMME STORAGE SECTION
EXECUTABLE PROGRAMME STORAGE SECTION
20C
20B
20A
EXECUTION

FIG.11

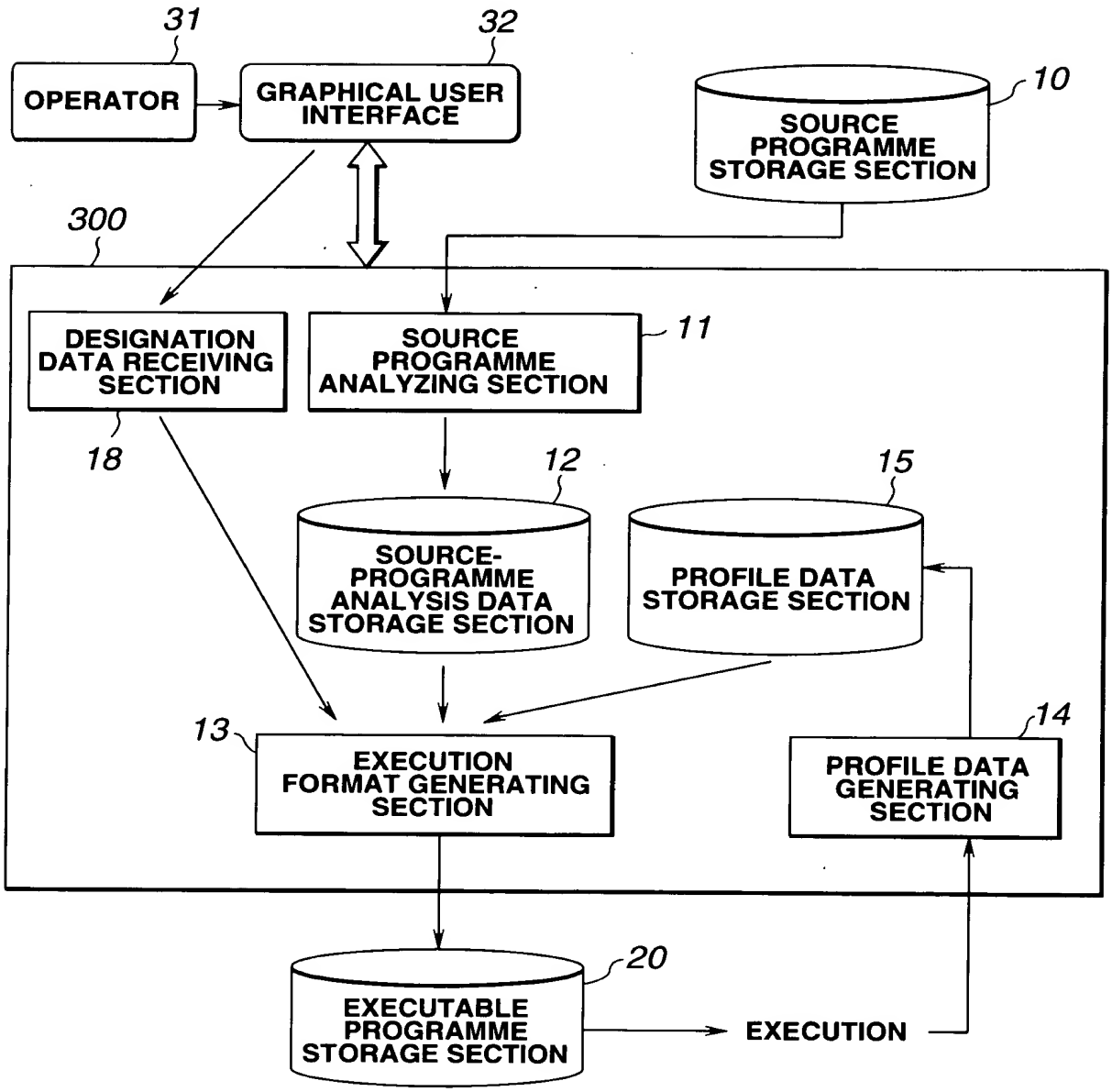

FIG.12

09486865 - 082799

09/486865

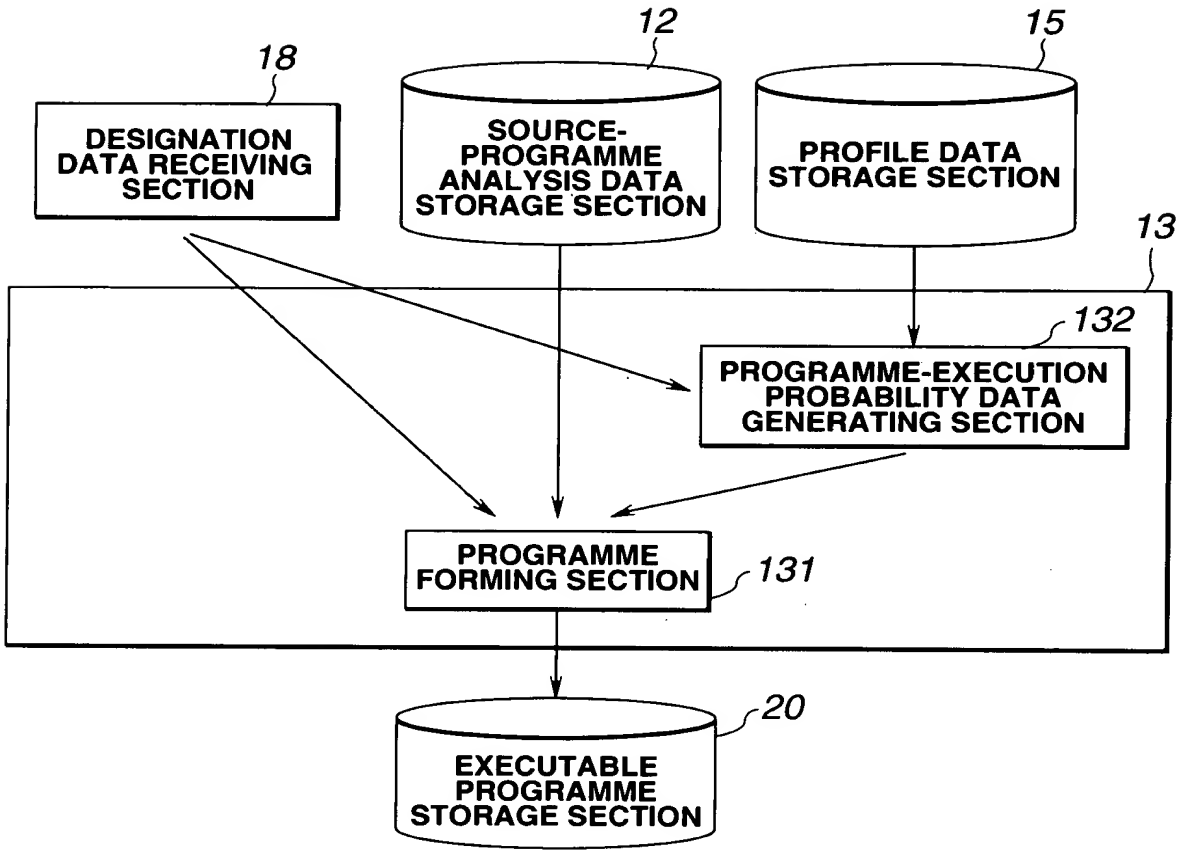

FIG.13

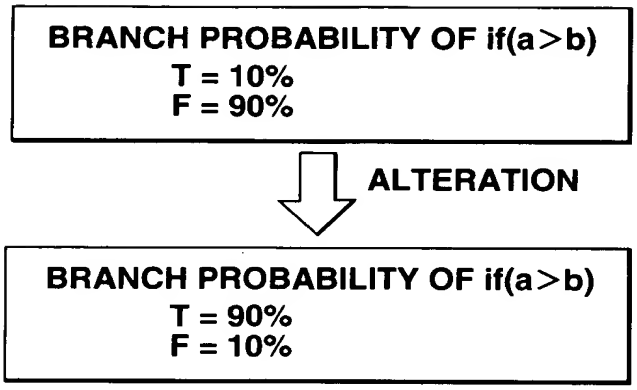

FIG.14

```
. text
block_1 :
        ld      r1, [a]         ; r1<-a
        ld      r2, [b]         ; r2<-b
        mv      r4, r1          ; r4<-r1
        cmp     r3, r1, r2      ; r3<-r1 cmp r2
        ble     r3, block_3
block_2 :
        ld      r4, [a]         ; c<-r4
        mul     r8, r4, r4      ; r8<-r4 * r4
        st      r8, [d]         ; d<-r8
        jmp     block_5
block_3 :
        st      r5, [b]         ; r5<-b
        st      r5, [c]         ; c<-r5
        jmp     block_4
block_4 :
        ld      r6, [c]         ; r6<-c
        ld      r7, [c]         ; r7<-c
        mul     r8, r6, r7      ; r8<-r6 * r7
        st      r8, [d]         ; d<-r8
```

FIG.15

514 Rec'd PCT/PTO 2 MAR 2000

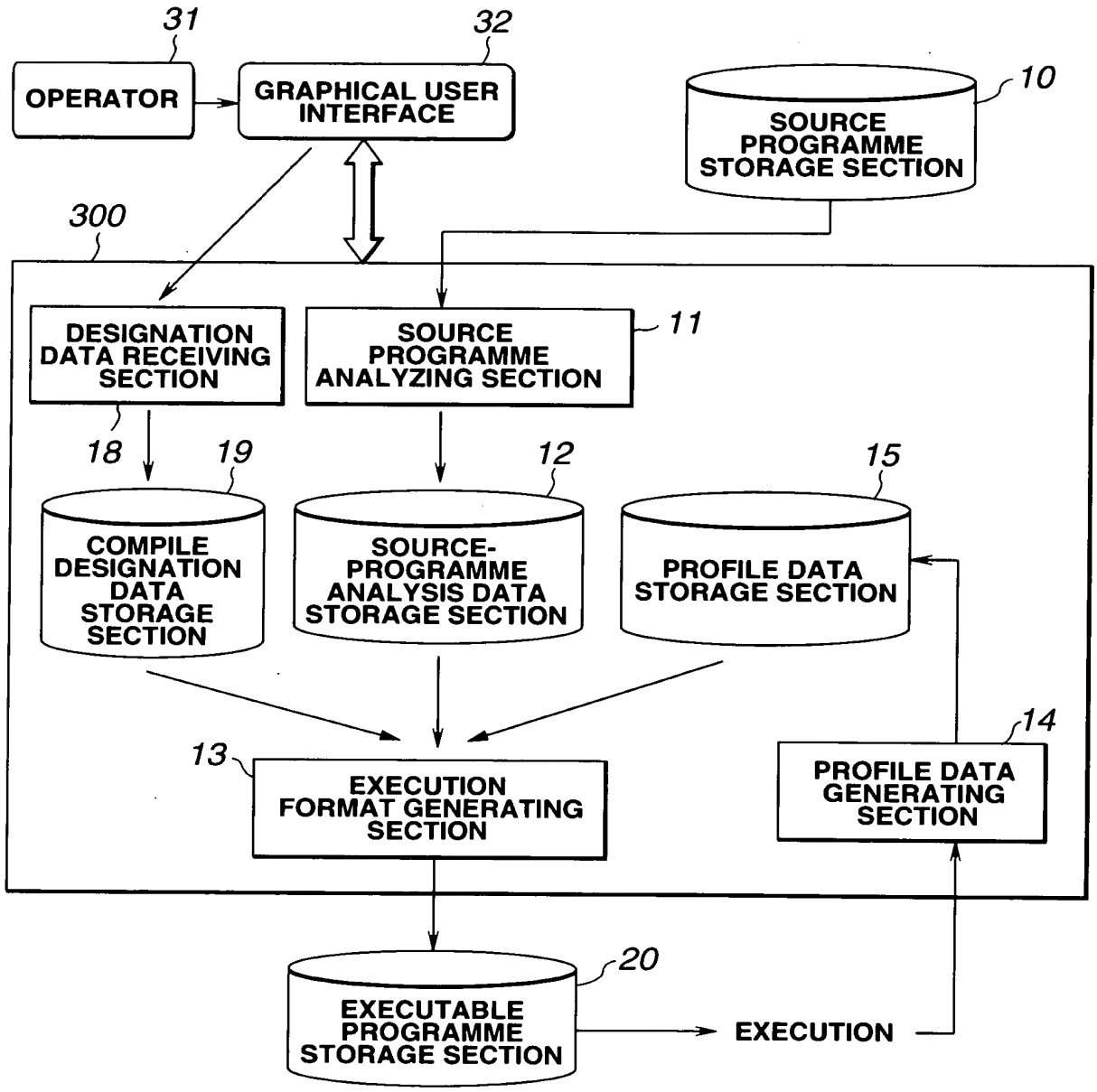

FIG.16